Simulasi *Vending Machine* Dengan Mengimplementasikan *Finite State Automata*

Tri Ichsan Saputra[1], Fauziah[2], Aris Gunaryati[3]
[1]triichsan1@gmail.com, [2]fauziah@civitas.unas.ac.id, [3]umidanti@gmail.com

Program Studi Teknik Informatika, Universitas Nasional

Abstract-In theory of language and automata explain about abstract machine which in there is finite state automata which can be implemented into vending machine. Vending machine for a city like Jakarta is not a difficult subject to find it. Operational way is already widely known procedures, but about how it works in the machine, still not many know it, through this simulation is expected the user also better understand it.

Keywords: Vending Machine, Simulation, Finite State, Language and Automata.

Abstrak-Pada teori bahasa dan automata menjelaskan tentang mesin-mesin abstrak yang mana didalamnya terdapat finite state automata yang bisa diimplementasikan kedalam vending machine. Vending machine untuk kota seperti Jakarta sudah bukan perihal sulit untuk menemukannya. Cara kerja secara operasional sudah banyak diketahui prosedurnya, namun perihal bagaimana cara kerjanya didalam mesin tersebut, masih belum banyak yang mengetahuinya, melalui simulasi ini diharapkan pengguna juga lebih paham akan hal itu.

Kata Kunci : Vending Machine, Simulasi, Finite State, Bahasa dan Automata.

I. PENDAHULUAN

Teori automata sangat erat kaitannya dengan mesin-mesin abstrak. Sedang teori bahasa erat kaitannya dengan komunikasi atau penghubung baik antara sesama manusia maupun dengan yang lainnya, jika diambil dalam materi ini maka bahasa adalah media antara manusia dengan komputer untuk saling berinteraksi[1].

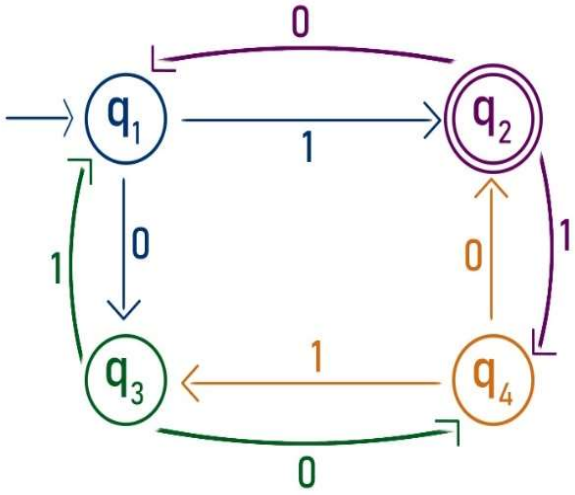
Dari jenisnya yaitu *finite state automata (FSA)* terdapat mesin bahasa yang berarti dapat mengenali, menerima dan *output* yang mana berbeda dengan jenis FSA sebelumnya (DFA & NFA), pada jenis ini tidak terdapat *state* menerima dan menolak, memiliki fungsi & himpunan *output.*

Sebagai contoh mesin FSA *output* bisa dengan menggunakan simulasi *vending machine* yang mana konsep dasar dari alat tersebut adalah metode FSA *output*. Sebuah *vending machine* biasanya menjual 1 (satu) jenis barang, seperti : makanan ringan (*snack*), minuman, pembelian tiket seperti halnya di KRL, dan sebagainya. Walaupun di Indonesia, khususnya Jakarta sudah banyak penggunaan *vending machine*, namun alat bantu simulasi akan membantu dalam mempelajari proses dari *vending machine* tersebut.

II. METODE PENELITIAN

Metode yang digunakan yaitu dengan menggunakan *finite state*, yang mana jika diambil contoh kasus sebagai berikut :

Tuple M pada FSA jenis DFA maupun NFA diantaranya (Q,Σ,δ,S,F) yang mengartikan untuk Q=himpunan *state*, Σ=himpunan *input*, δ=fungsi transisi, S=*state* awal, F=*state* akhir. Disini sebagai contoh sederhana, kita gunakan FSA jenis DFA[3].



Gambar 1: Diagram DFA

Dari diagram diatas, kita bisa melihat bahwa tuplenya sebagai

Q = (q1, q2, q3, q4), Σ = (0, 1), S=(q1), F=(q2)

Fungsi transisi=δ

δ(q1,0)=q3, δ(q1,1)=q2, δ(q2,0)=q1, δ(q2,1)=q4, δ(q3,0)=q4, δ(q3,1)=q1, δ(q4,0)=q2, δ(q4,1)=q3

View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
Journal of Information Technology and Computer Science (JOINTECS)

Dari fungsi transisi tersebut maka dapat dilihat tabel transisi sebagai berikut :

Table 1: Tabel Transisi

δ	0	1
q1	q3	q2
q2	q1	q4
q3	q4	q1
q4	q2	q3

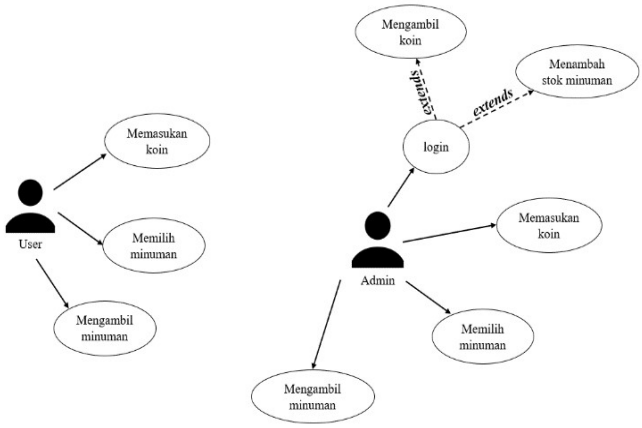
Sebagai contoh, misal kita *input* '1111', maka untai akan bergerak dari *state* awal (q1) kemudian *input* '1' pertama berubah posisi menjadi (q2) lalu *input* kembali '1' maka posisi pindah menjadi (q4), kemudian *input* kembali '1', maka posisi akan pindah kembali menjadi (q3), dan input kembali '1' yang terakhir, maka posisi akan berakhir pada (q0) dan hasilnya akan ditolak karena tidak berakhir di *state* penerima / *finish state*.

Kasus diatas sudah menggambarkan sebagian kecil dari proses yang ada didalam *vending machine*, sama halnya seperti kasus diatas, nantinya jika nilai *input* (uang) kurang dari harga yang ada pada *vending machine*, maka *output* (minuman) tidak akan keluar.

Perancangan Sistem

Sistem dirancang dengan menggunakan UML (*Unified Modelling Language*) yang terdiri dari *use case diagram dan activity diagram*[9].

Usecase diagram mendeskripsikan sistem dari sudut pandang *user*, *usecase diagram* digunakan untuk melihat secara visual bagaimana simulasi perwujudan atau interaksi kejadian yang terjadi antara pengguna dengan *vending machine* dan apa yang dapat dilakukan oleh *user*, *user* dibagi menjadi dua *user*, yaitu *user* itu sendiri dan *admin*.

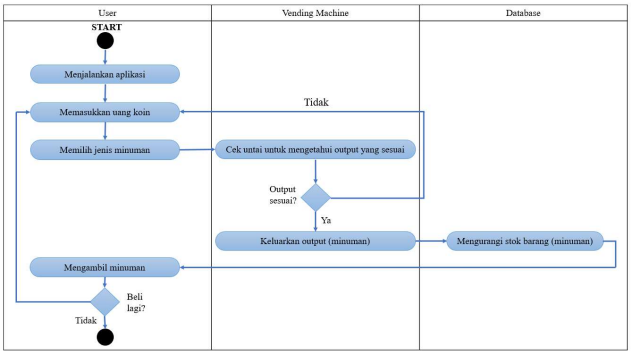


Gambar 2: ***(a)*** *Usecase diagram user* ***(b)*** *Usecase diagram admin*

Usecase diagram user dapat dilihat pada gambar 2(a), setelah menjalankan aplikasi, *user* dapat melakukan hal sebagai berikut; memasukan koin, memilih minuman dan mengambil minuman.

Kemudian dalam *usecase diagram admin* yang ada pada gambar 2(b), *admin* dapat melakukan hal yang dapat dilakukan oleh *user* dan juga dapat *login* untuk melakukan hal perihal mengambil koin atau menambah stok minuman.

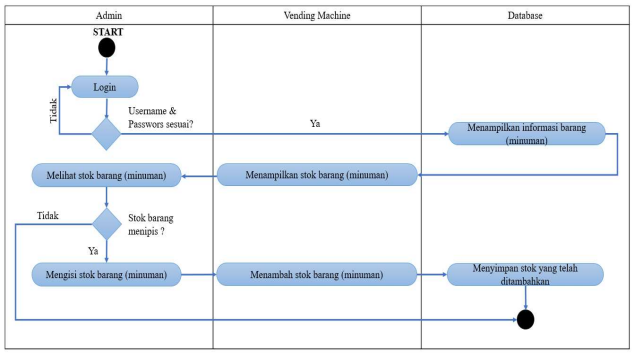
Activity diagram menggambarkan sebuah alur dari rangkaian kegiatan yang ada pada sistem yang sedang dirancang. Jika *vending machine* dijalankan oleh *user*, maka *activity diagram* akan tergambar seperti gambar 3.



Gambar 3: Activity Diagram User

User mulai menjalankan aplikasi, kemudian memasukkan uang dan memilih jenis minuman yang diinginkan, kemudian sistem akan mengecek dan menyamakan antara *input* (uang) dan *output* (minuman) yang dipilih, jika sudah sesuai, maka *vending machine* akan mengeluarkan *output* (minuman) sesuai keinginan dan *database* akan mengurangi stok minuman, lalu tahap terakhir, *user* mengambil minuman yang telah dikeluarkan *vending machine*. Dan jika *user* ingin membeli lagi, alurnya akan terulang mulai dari memasukkan koin ke *vending machine*.

Kemudian *activity diagram admin* dapat dilihat pada gambar 4.



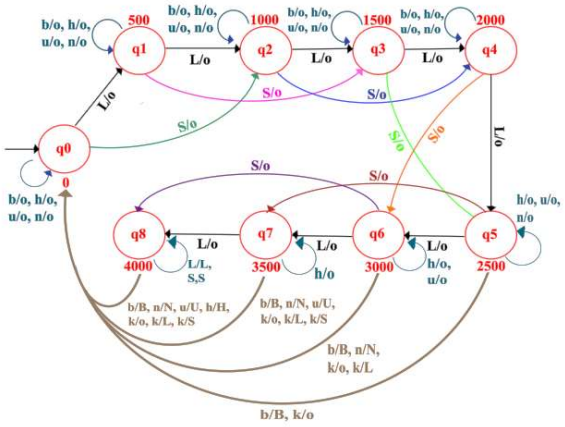
Gambar 4: Activity Diagram Admin

Activity diagram admin berbeda, disini *activity diagram* tergambar perihal mengisi ulang stok minuman. Dimulai dari *login*, jika *username* dan *password* tidak sesuai

maka harus mengulang hingga *username* dan *password* sesuai, kemudian *database* akan menampilkan informasi minuman yang akan tampil didalam *vending machine*, kemudian *admin* melihat stok minuman, jika dirasa telah menipis maka ditambahkan, dan jika tidak, *activity diagram* selesai sampai disitu.

Perancangan Diagram State

Untuk merancang simulasi ini dibutuhkan *diagram state* yang digunakan untuk mengetahui cara kerja dari simulasi *vending machine* ini, semua nominal dengan minuman tidak sesuai hanya dibuat perumpamaan untuk mendeskripsikannya.



Gambar 5: Diagram State FSA Output

Gambar 5 menjelaskan bahwa terdapat beberapa input, sebagai berikut : k (kembalian), L (logam 500), S (logam 1000), b (pilih Buavita), n (pilih Nescafe), u (pilih Ultra Milk), H (pilih Hilo). Dan dengan output, diantaranya : o (tidak ada), B (keluar Buavita), N (keluar Nescafe), U (keluar Ultra Milk), H (keluar Hilo).

Pendefinisian Tuple

FSA *Output* didefinisikan sebagai berikut:

M=(Q,Σ,δ,S,Δ,λ)

Yang mana:

Q=himpunan *state*, Σ=himpunan simbol *input*, δ=fungsi transisi, S=*state* awal, Δ=himpunan simbol *output*, λ=fungsi *output.*

Sehingga dapat didefinisikan sebagai berikut:

Q={q0, q1, q2, q3, q4, q5, q6, q7, q8}

Σ={b,n,u,h,L,S,k}

δ(q0,b)=q0; δ(q0,n)=q0; δ(q0,u)=q0; δ(q0,h)=q0; δ(q0,L)=q1; δ(q0,S)=q2; δ(q0,k)=q0;

δ(q1,b)=q1; δ(q1,n)=q1; δ(q1,u)=q1; δ(q1,h)=q1; δ(q1,L)=q2; δ(q1,S)=q3; δ(q1,k)=q1;

δ(q2,b)=q2; δ(q2,n)=q2; δ(q2,u)=q2; δ(q2,h)=q2; δ(q2,L)=q3; δ(q2,S)=q4; δ(q2,k)=q2;

δ(q3,b)=q3; δ(q3,n)=q3; δ(q3,u)=q3; δ(q3,h)=q3; δ(q3,L)=q4; δ(q3,S)=q5; δ(q3,k)=q3;

δ(q4,b)=q4; δ(q4,n)=q4; δ(q4,u)=q4; δ(q4,h)=q4; δ(q4,L)=q5; δ(q4,S)=q6; δ(q4,k)=q4;

δ(q5,b)=q0; δ(q5,n)=q5; δ(q5,u)=q5; δ(q5,h)=q5; δ(q5,L)=q6; δ(q5,S)=q7; δ(q5,k)=q0;

δ(q6,b)=q0; δ(q6,n)=q0; δ(q6,u)=q6; δ(q6,h)=q6; δ(q6,L)=q7; δ(q6,S)=q8; δ(q6,k)=q0;

δ(q7,b)=q0; δ(q7,n)=q0; δ(q7,u)=q0; δ(q7,h)=q7; δ(q7,L)=q8; δ(q7,S)=q8; δ(q7,k)=q0;

δ(q8,b)=q0; δ(q8,n)=q0; δ(q8,u)=q0; δ(q8,h)=q0; δ(q8,L)=q8; δ(q8,S)=q8; δ(q6,k)=q0;

Jika dipetakan dalam tabel transisi, seperti tabel 2.

Tabel 2:Tabel transisi input

δ	b	n	u	h	L	S	k
q0	q0	q0	q0	q0	q1	q2	q0
q1	q1	q1	q1	q1	q2	q3	q1
q2	q2	q2	q2	q2	q3	q4	q2
q3	q3	q3	q3	q3	q4	q5	q3
q4	q4	q4	q4	q4	q5	q6	q4
q5	q0	q5	q5	q5	q6	q7	q0
q6	q0	q0	q6	q6	q7	q8	q0
q7	q0	q0	q0	q7	q8	q8	q0
q8	q0	q0	q0	q0	q8	q8	q0

Tabel 2 menjelaskan tentang perpindahan state sesuai dengan *input* yang masuk.

S={q0}

Δ={B,N,U,H,o}

λ=fungsi *output* untuk setiap transisi

λ(q0,b)=o; λ(q0,n)=o; λ(q0,u)=o; λ(q0,h)=o; λ(q0,L)=o; λ(q0,S)=o; λ(q0,k)=o;

λ(q1,b)=o; λ(q1,n)=o; λ(q1,u)=o; λ(q1,h)=o; λ(q1,L)=o; λ(q1,S)=o; λ(q1,k)=o;

λ(q2,b)=o; λ(q2,n)=o; λ(q2,u)=o; λ(q2,h)=o; λ(q2,L)=o; λ(q2,S)=o; λ(q2,k)=o;

λ(q3,b)=o; λ(q3,n)=o; λ(q3,u)=o; λ(q3,h)=o; λ(q3,L)=o; λ(q3,S)=o; λ(q3,k)=o;

λ(q4,b)=o; λ(q4,n)=o; λ(q4,u)=o; λ(q4,h)=o; λ(q4,L)=o; λ(q4,S)=o; λ(q4,k)=o;

λ(q5,b)=B; λ(q5,n)=o; λ(q5,u)=o; λ(q5,h)=o; λ(q5,L)=o; λ(q5,S)=o; λ(q5,k)=o;

λ(q6,b)=B; λ(q6,n)=N; λ(q6,u)=o; λ(q6,h)=o; λ(q6,L)=o; λ(q6,S)=o; λ(q6,k)=o,L;

λ(q7,b)=B; λ(q7,n)=N; λ(q7,u)=U; λ(q7,h)=o; λ(q7,L)=o; λ(q7,S)=o; λ(q7,k)=o,L,S;

λ(q8,b)=B; λ(q8,n)=N; λ(q8,u)=U; λ(q8,h)=H; λ(q8,L)=L; λ(q8,S)=S; λ(q8,k)= o,L,S;

Jika dipetakan dalam tabel transisi, seperti tabel 3.

Tabel 3: Tabel transisi output

δ	b	n	u	h	L	S	k
q0	o	o	o	o	o	o	o
q1	o	o	o	o	o	o	o
q2	o	o	o	o	o	o	o
q3	o	o	o	o	o	o	o
q4	o	o	o	o	o	o	o
q5	B	o	o	o	o	o	o
q6	B	N	o	o	o	o	o,L
q7	B	N	U	o	o	o	o,L,S
q8	B	N	U	H	L	S	o,L,S

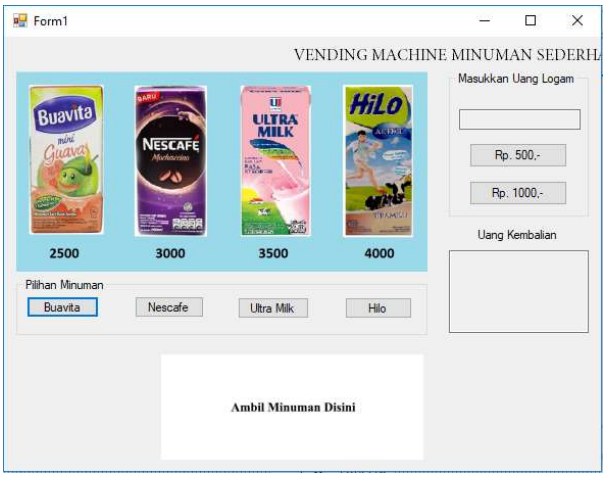
Dengan tabel 3 tersebut, kita dapat mengetahui *input* yang akan dikeluarkan, sebagai contoh pada *state* q8 yang mengeluarkan banyak pilihan *output*, jika mendapatkan *input* b, maka akan keluar *output* B (Buavita), jika mendapatkan *input* n, maka akan keluar *output* N (Nescafe), jika mendapatkan *input* u, maka akan keluar *output* U (Ultra Milk), jika mendapatkan *input* h, maka akan keluar *output* H (Hilo). Bila mendapat *input* L (500), maka *output*-nya adalah L lagi, dan bila *input*nya S (1000), *output*-nya juga S. *State* q8 adalah *state* terakhir, sehingga bila diberi *input* nilai uang lagi (L dan S), maka *state* tidak akan beranjak dan memuat nilai kelebihan uang tersebut. Bila *input*nya adalah k (kembalian), maka kelebihan dari L dan S tadi akan dikeluarkan sesuai dengan nilainya.

Output: Dari hasil kalkulasi, diperoleh *output* yang akan dikeluarkan aplikasi. *Output* berupa gambar minuman yang dipilih, serta angka yang menyatakan uang kembalian (jika nominal uang lebih besar dari harga yang tertera).

III. HASIL DAN PEMBAHASAN

Form Simulasi Vending Machine

Rancangan tampilan *form vending machine* dapat dilihat pada gambar 6, simulasi tersebut sudah memvisualisasikan gambaran dari *vending machine* tersebut yang memfasilitasi untuk pembelian minuman [4].

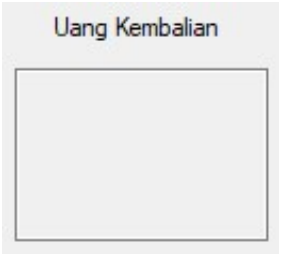


Gambar 6: Simulasi Vending Machine

Pada gambar 7 dapat dilihat terdapat fasilitas memasukkan uang, hanya ada uang Rp. 500 dan Rp. 1000 dikarenakan *input* berupa koin (uang logam), jika kita *input* Rp. 500 maka dilayar akan tercetak nominal yang sama dengan apa yang di*input*kan, kemudian jika kita melakukan *input* selanjutnya, nominal akan berubah (bertambah) sesuai dengan nominal yang ditambahkan.

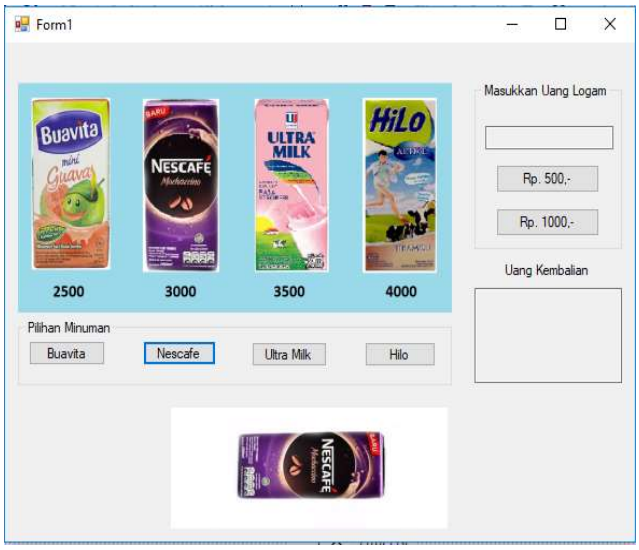
Gambar 7: Fasilitas Input Koin

Kemudian pada gambar 8 dapat dilihat ada fasilitas uang kembalian, hal itu bisa terjadi jika nominal yang dimasukkan lebih besar dari harga minuman yang dipilih, dan akan kosong jika nominal uang yang dimasukkan sama atau bahkan kurang dari harga minuman yang dipilih.



Gambar 8: Fasilitas Uang Kembalian

Sebagai contoh, jika kita ingin membeli minuman Nescafe harga Rp. 3.000,- maka akan muncul tampilan seperti gambar 9.



Gambar 9: Hasil Keluaran dari Vending Machine

Minuman Nescafe bisa keluar dikarenakan kita telah menginput nominal yang sesuai dengan harga minuman tersebut, dan tidak ada uang kembalian dikarenakan harga yang dimasukkan sama dengan dan atau tidak melebihi dari harga minuman. Hal ini menunjukkan sistem telah sesuai dengan rancangan FSA [5]

IV KESIMPULAN

Penggunaan *finite state automata* pada *vending machine* sudah sangat sesuai didalam cara kerjanya, *finite state* sebagai dasar dari pengoperasian simulasi tersebut sudah menjelaskan *detail* alur dari apa yang *user* lakukan, sehingga prosesnya mudah dipahami. Dan diharapkan *user* dapat lebih paham akan hal tersebut.

DAFTAR PUSTAKA

[1] Widyasari, "Telaah Teoritis Finite State Automata Dengan Pengujian Hasil Pada Mesin Otomata." Jurnal Ilmiah SISFOTENIKA, Vol. 1, No. 1, Januari 2011

[2] Sinurat, S. SIMULASI TRANSFORMASI REGULAR EXPRESSION TERHADAP FINITE STATE AUTOMATA. *Pelita Informatika: Informasi dan Informatika*, *4*(1).2013

[3] Suparyanto, Selo, Simulator Pengenal String Yang Diterima Sebuah Deterministic Finite Automata (DFA), CITEE, pp. 377–381.2017

[4] Wamiliana, W., Kurniawan, D., & Melly, R. I. Penerapan Konsep Finite State Automata (FSA) pada Mesin Pembuat Minuman Kopi Otomatis. *Jurnal Komputasi*, *1*(1). 2016

[5] Irawan, J. C., Pakereng, I. M., & Somya, R. (2012). Perancangan dan Implementasi Finite Automata pada Simulasi Vending Machine. *de CARTESIAN*, *1*(1), 42-52.